

# Homework 5

CMSY-217, Spring 2013

The source code for this assignment must be submitted using the Canvas course website prior to the start of class on Thursday, April 25.

The Social Security Administration provides a webpage (<http://www.ssa.gov/OACT/babynames>) with two interactive applications that allow you to display the most popular baby names for a selected year and track the popularity of a selected baby name over several years. The data are based on Social Security Card applications for years 1880 through 2011.

Popularity in 2011			Popularity of the female name Chloe		
Rank	Male name	Female name	Year of birth		Rank
1	Jacob	Sophia	2011		10
2	Mason	Isabella	2010		9
3	William	Emma	2009		9
4	Jayden	Olivia	2008		10
5	Noah	Ava	2007		16
6	Michael	Emily	2006		18
7	Ethan	Abigail	2005		19
8	Alexander	Madison	2004		23
9	Aiden	Mia	2003		24
10	Daniel	Chloe	2002		25
Note: Rank 1 is the most popular, rank 2 is the next most popular, and so forth.			2001		30
			2000		38
			1999		63
			Note: Rank 1 is the most popular, rank 2 is the next most popular, and so forth. Name data are from Social Security card applications for births that occurred in the United States.		

The data have been placed in a Java DB (Apache Derby) database named **babynames** with a table for each year of birth that contains columns for name, sex, and number of births. Each table is named with the letters **YOB** followed by the four-digit year for which the data was collected. For example, the table **YOB2011** contains the data for year 2011. Note that each name entry in these tables begins with a capital letter and is followed by all lowercase letters.

```
ij> DESCRIBE YOB2011;
```

COLUMN_NAME	TYPE_NAME
NAME	VARCHAR
SEX	CHAR
NUMBER	INTEGER

There is also a table called `TOTALBIRTHS` which contains the total number of births, by gender, for each year.

```
ij> DESCRIBE TOTALBIRTHS;
COLUMN_NAME          | TYPE_NAME
-----
BIRTHYEAR             | INTEGER
MALE                  | INTEGER
FEMALE                | INTEGER
```

The `BabyNames` class contains a Swing application that provides you with a graphical user interface (GUI) similar in appearance to the HTML forms on the Social Security website. In addition, the event-handling code has been written so that when the user clicks the `Go` button - a `BabyNamesQuery` object is created, the input parameters are passed to the `getList` or `getRank` method, and a results `String` is returned which is displayed in the `JTextArea` at the bottom of the GUI. The following figure shows the `BabyNames` application running with the results of a `getList` method call displayed.

The screenshot shows a Java Swing window titled "Baby Names". It contains two main panels at the top and a large text area at the bottom.

**Popular Names by Birth Year**  
For a list of the most popular names for a particular year of birth (any year after 1879), enter the year and the length of the popularity list.

Enter year of birth:    
Popularity:    
Name ranking may include:  
☐ Percent of total births  
☐ Number of births  
☒ Neither

**Popularity of a Name**  
See how popularity of a name has changed over time!

Name?    
  
Do not use spaces, hyphens, or other non-alphabetic characters in the name.  
Sex associated with name  
☐ Male ☐ Female ☒ None  
Number of years?

**Popularity in 2011**

Rank	Male Name	Female Name
1	Jacob	Sophia
2	Mason	Isabella
3	William	Emma
4	Jayden	Olivia
5	Noah	Ava
6	Michael	Emily
7	Ethan	Abigail
8	Alexander	Madison
9	Aiden	Mia
10	Daniel	Chloe
11	Anthony	Elizabeth
12	Matthew	Ella
13	Elijah	Addison
14	Joshua	Natalie
15	Liam	Lily
16	Andrew	Grace
17	James	Samantha
18	David	Avery
19	Benjamin	Sofia
20	Logan	Aubrey

1. Write the `getList` method in the `BabyNamesQuery` class using JDBC to provide the same functionality as the **Popular Names by Birth Year** application on the Social Security website.
2. Write the `getRank` method in the `BabyNamesQuery` class using JDBC to provide the same functionality as the **Popularity of a Name** application on the Social Security website.

The following SQL statements are examples of the `String` query objects that you could pass to the `executeQuery` method of the `Statement` interface to return a `ResultSet` object. Figure 28.23 from the textbook would be a good starting point for the necessary Java code.

```
SELECT NAME
FROM YOB2011
WHERE SEX='M'
ORDER BY NUMBER DESC, NAME ASC
FETCH FIRST 20 ROWS ONLY
```

```
SELECT NAME, NUMBER
FROM YOB2011
WHERE SEX='M'
ORDER BY NUMBER DESC, NAME ASC
FETCH FIRST 20 ROWS ONLY
```

```
SELECT MALE
FROM TOTALBIRTHS
WHERE BIRTHYEAR=2011
```

```
SELECT NUMBER
FROM YOB2011
WHERE NAME='Chloe' AND SEX='F'
```

```
SELECT COUNT (NAME)
FROM YOB2011
WHERE SEX='F' AND NUMBER > 10917
OR SEX='F' AND NUMBER = 10917 AND NAME<='Chloe'
```